UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

OPTIMAL URBAN BUS ROUTING WITH SCHEDULING FLEXIBILITIES

by

François Soumis, Jacques Desrosiers
and Martin Desrochers

This research will appear in the Proceedings of the 11th IFIP Conference on SYSTEM MODELLING AND OPTIMIZATION, Ed. P. Thoft-Christensen, Copenhagen, Denmark, July 25-29, 1983.

The authors are grateful to the Canadian National Sciences and Engineering Research Council and to the research department of Ecole des Hautes Etudes Commerciales de Montréal for their financial support.

Université de Montréal

Centre de recherche sur les transports - Publication #285
Ecole des Hautes Etudes Commerciales
- Cahiers du GERAD #G-83-33

Novembre 1983

Dépôt légal - 1er trimestre 1984

OPTIMAL URBAN BUS ROUTING WITH SCHEDULING FLEXIBILITIES

F. Soumis, J. Desrosiers, M. Desrochers
Ecole des Hautes Etudes Commerciales de Montréal
Montréal, Canada H3T 1V6

1. INTRODUCTION

Bus fleet route planning is often carried out in the following two sequential stages:

1) Based on the demand, determine the trips to be carried out.
2) Assign buses to the trips so as to minimize total costs.

Step 1 fixes the starting time of each trip taking into account the demand, but without considering bus assignment. Step 2 optimizes bus assignment without modifying the trip starting times established in step 1. The resulting operating plan is suboptimal: costs can be reduced without significantly affecting the quality of service by slightly modifying the schedules of certain trips a posteriori to reduce the number of vehicles required and the total travelling time.

We propose to fix the departure times during cost minimization in step 2. Step 1 involves only the determination of an interval during which each trip must begin to ensure an adequate quality of service.

The bus assignment problem with fixed departure times can be formulated as a minimum cost flow problem whose optimal solution is easily obtained [8]. With flexible departure times, the problem becomes more difficult. Good solutions have been obtained within acceptable computation times using heuristic methods. One method which is particularly suitable for this type of problem is that developed by Bokinge and Hasselstrom [3] and integrated into the Volvo Traffic Planning Package. The optimal method proposed here is capable of solving problems encountered in practice. Results for problems with 128 and 158 departures are presented in section 10. The problems come from two Swedish cities and were suggested to us by the authors of [3].

We now introduce the terminology used. A <u>trip</u> is an itinerary which must be carried out by the same bus. Trip i is characterized by an origin, a destination, a duration, a cost and a time interval $[a_i, b_i]$ during which the trip must begin. An <u>intertrip</u> arc is an empty run which may be carried out by a vehicle. Intertrip arc (i,j) goes from the destination of trip i to the origin of trip j. Its duration t_{ij} and its cost c_{ij} include respectively the duration and cost of trip i. <u>A route</u> is a sequence of trips and intertrip arcs carried out by a vehicle between two visits to the depot. A <u>route block</u> is a sequence of routes carried out by the same vehicle during the day.

2. MATHEMATICAL FORMULATION

The buses flow through a network made up of a set of nodes representing the trips $i = 1,\ldots,n$ and the depot (node 0) and joined by a set of arcs which include the intertrip arcs (i,j), $i,j = 1,\ldots,n$, arcs $(0,j)$ $j = 1,\ldots,n$ joining the depot to the origin of each trip, and arcs $(i,0)$, $i = 1,\ldots,n$ joining the destination of each trip to the depot. The structure of the network varies depending on whether there is only one or several vehicle depots, and whether vehicles visit the depot only once or several times in a day. The network with multiple exits from a single depot is presented in section 4.

The mathematical formulation includes two types of variables: flow variables x_{ij} taking the value 1 when arc (i,j) is used by a vehicle, and continuous time variables t_i, associated with the departure time of each trip. The variable travel costs are associated with the flow variables corresponding to the intertrip arcs, and a fixed cost per vehicle is associated with the flow variables corresponding to the depot exit arcs.

The optimal routes respecting the scheduling constraints are the solution of the following problem:

$$\text{Min} \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ij} + W \sum_{i=0}^{n} \sum_{j=0}^{n} x_{ij}(t_j - t_{ij} - t_i) \qquad (1)$$

$$\sum_{j=0}^{n} x_{ij} = \sum_{i=0}^{n} x_{ji} = 1 \qquad i=1,\ldots,n \qquad (2)$$

$$x_{ij} \geq 0 \qquad i,j=0,1,\ldots,n \qquad (3)$$

$$x_{ij} > 0 \Rightarrow t_i + t_{ij} \leq t_j \qquad i,j=1,\ldots,n \qquad (4)$$

$$a_i \leq t_i \leq b_i \qquad i=1,\ldots,n \qquad (5)$$

$$x_{ij} \text{ binary} \qquad i,j=0,1,\ldots,n \qquad (6)$$

The first term in the objective function includes the fixed costs of the vehicles and the travel costs while the second term evaluates the cost of waiting between trips. (W is the cost of a one minute wait). Note that the waiting cost term is non-linear but becomes linear if the schedule is fixed.

If the waiting costs are dropped, relations (1), (2) and (3) form a routing problem without scheduling constraints. This is a minimum cost flow problem which is easily solved, and whose solution is integer. Relations (4) describe the compatibility requirements between the routes and the schedule. Constraints (5) establish the

time intervals within which trips must begin.

3. LITERATURE REVIEW

One possible heuristic solution method involves the discretization of the time intervals and the replacement of each variable t_i by a set of binary variables associated with each decision to begin trip i or not at a discrete point in time. Levin [10] used this approach for an air transportation problem and Swersey and Ballard [11] used it for school transportation; they found that an optimal integer solution is often obtained when using the simplex algorithm on the problem without integrality constraints. Note, however, that these authors restricted the objective function to deal with the number of vehicles only, after discovering that it was more difficult to obtain integrality when travel costs were also included. Other methods suggested by Bodin et al [2] include travel costs but additional approximations have to be made.

Bokinge and Hasselstrom [3] have developed a heuristic network algorithm for the problem which involves the solution of several fixed schedule problems using the minimum cost flow algorithm. An initial fixed schedule problem considers the "nucleus" of each trip in order to identify peak periods and to obtain a lower bound for the number of vehicles. The trip "nucleus" is defined by fixing the departure time for the trip at the end of the starting time interval, and by fixing the ending time as if the trip began at the beginning of the time interval. The authors then continue with the original trips fixing the schedules so as to increase the chance of obtaining good routes. The fixed schedule problem is handled using the minimum cost flow algorithm. The schedule is then modified and the solution process is repeated until a satisfactory solution is obtained.

Branch-and-bound approaches are the most commonly used methods for the optimal solution of this type of problem. A relaxed problem is solved and branching is carried out to eliminate infeasible solutions. Several relaxations of the problem are possible; each represents a different compromise between solution cost and quality of approximation.

A very good approximation was obtained with the simplex algorithm by relaxing the integrality of the covering problem whose columns are routes satisfying the scheduling constraints [6]. To avoid enumerating all possible routes, a column generation procedure is used in which the subproblems to be solved are shortest path problems with scheduling constraints [5]. Excellent results were obtained with this approach in the single depot case with only one exit from the depot per vehicle and where vehicle waiting time was not considered. On the other hand, this approach required $200\ 000_8$ words of central memory to handle a problem with 151 trips.

An approximation which is easier to solve is obtained by relaxing constraints (4) and (5) and by suppressing the waiting time term; an integer assignment problem is obtained and this is easily solved without excessive memory requirements. This relaxation has been successfully used by Carpaneto and Toth [4] for the travelling salesman problem - a special case of the problem studied here. The present authors [7] have also used this approach to handle this problem in the case of a single exit per vehicle from the depot, and no waiting costs. In this article, we present an adaptation of this method to handle the more general case of multiple exits from the depot and an objective function which includes waiting costs as well as empty running costs and fixed vehicle costs.

4. THE NETWORK

For the problem with multiple exits from a single depot, intertrip arcs (i,j) are of two types: direct, or via the depot. A direct intertrip arc exists if

$$a_i + t_{ij} \leq b_j \ . \qquad (7)$$

Its cost includes the travel cost c_{ij} and the waiting cost $W(t_j - t_{ij} - t_i)$. An intertrip arc via the depot is possible if

$$a_i + t_{i0} + t_{0j} \leq b_j \ . \qquad (8)$$

Its cost is $c_{i0} + c_{0j}$ as there is no waiting cost at the depot. Note that in general, distances are Euclidean ($t_{i0} + t_{0j} \geq t_{ij}$) and that a direct intertrip journey is always possible if one via the depot is possible. When the two types of movement are possible only the one with the least cost is retained as an intertrip arc (i,j). In urban transit problems where the workday considered is much longer than individual trip length, the number of arcs increases rapidly as the number of trips increases. A morning trip can be followed by any afternoon trip, so the graph is almost complete and the number of intertrip arcs is of the order of $\frac{1}{4}n^2$. The networks obtained with the 128 trip and 158 trip problems had 6812 and 8073 arcs respectively. A reduced network (Figure 1) is obtained by replacing the arcs passing through the depot by a new set of arcs and nodes as defined below:

1- For each node i which has an intertrip arc leading from it via the depot, define a depot node with an associated time of $a_i + t_{i0}$, and an arc leading from i to the new node.

2- For each node j which has an intertrip arc via the depot arriving at it, define a depot node at time $b_j - t_{0j}$ and an arc from the new node to j.

3- Construct the sequence $\{N_k\}$ $k = 1,\ldots,K$ by classifying all the depot nodes in increasing time order ($K \leq 2n$).

4- Replace the old depot node by N_0 at the beginning of the period and by N_{K+1} at the end. Drop the 2n incident arcs.

5- Define the arcs (N_k , N_{k+1}) for $k = 0,\ldots,K$.

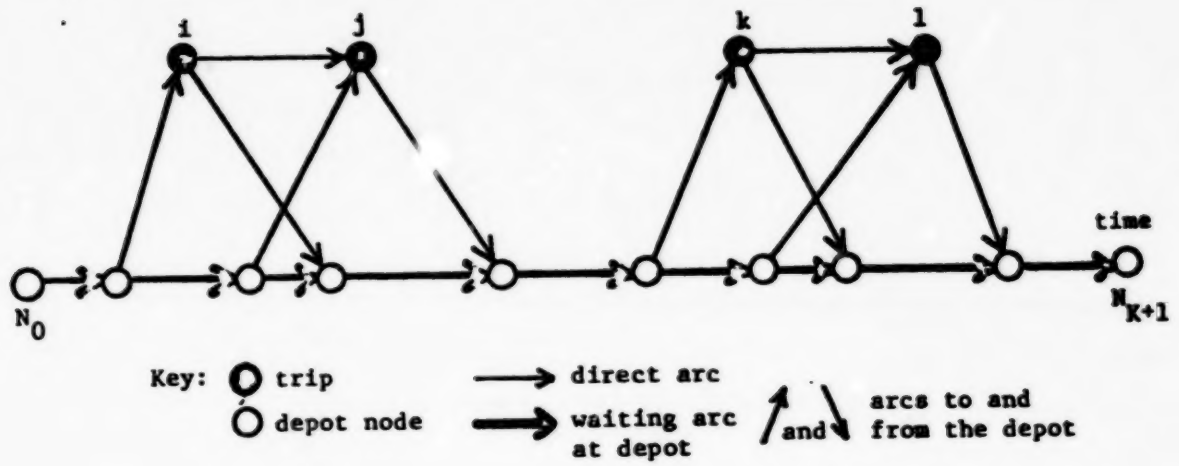

Figure 1: The network with multiple depot exits

This procedure replaces the 2n arcs incident to the depot and all the intertrip arcs via the depot (of the order of $\frac{1}{2}n^2$) by a maximum of 2n nodes and 3n arcs. For the 128 and 158 trip problems the reduced networks have 979 and 1320 arcs respectively. Note also that this reduced network retains all the essential information: there exists a path from i to j following the direction of the arcs in the reduced network if and only if there is such a path in the initial network.

5. SOLUTION OF THE RELAXED PROBLEM

Constraints (4) and (5) are relaxed so as to retain only the network constraints (2) and (3). Furthermore the waiting cost which depends on both the time variables and the flow variables is replaced by a lower bound which is a linear function of the flow variables only. This bound is $Wb_{ij}\ x_{ij}$ where $b_{ij} = \max\{0,\ a_j - b_i - b_{ij}\}$. Figure 2 illustrates the relationship between b_{ij} and the exact waiting time.

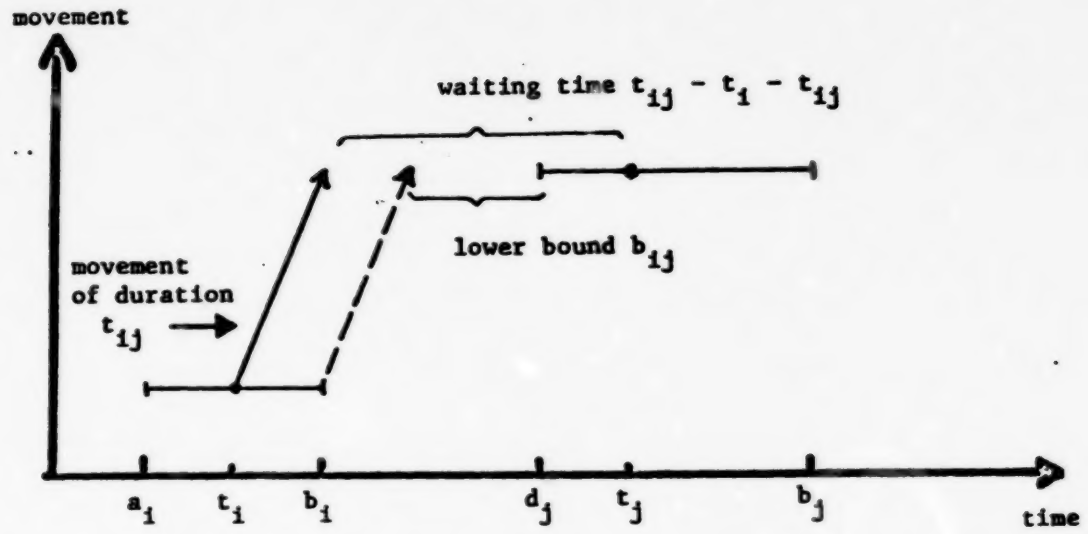

Figure 2: Lower bound on waiting time used in the network relaxation

A network problem is thus obtained and constraint (6) can be dropped as the problem will have an integer solution. This problem was solved using the RNET software [9] which is a version of the simplex algorithm designed especially for networks.

Note that the flows on the arcs incident to the trips are binary but that the flows on 2 consecutive depot nodes are integer. These flows represent the number of buses waiting at the depot at each time. To minimize the fixed cost associated with fleet size as well as travel and waiting costs, it is sufficient to add the fixed cost per vehicle to the arc (N_0 , N_1) with the variable x_{N_0,N_1} representing the number of buses used.

The solution of this problem gives a lower bound on the fixed costs, travel costs and waiting costs for the original problem.

6. TRIP SCHEDULES

In order to obtain a solution to the original problem from the preceding network solution, a schedule must be defined.

Note that in the network solution, all the trips are on unitary flow routes beginning at a depot node N_r, using an exit arc, a sequence of direct intertrip arcs and a depot entrance arc leading to a depot node N_s. The duration of each route is first determined, along with the time interval within which it must begin to be feasible with minimum waiting time. For a route of v trips we define iteratively for each subroute made up of a depot node N_r and trips $1,2,\ldots,k (k \leq v)$:

$-\ d^k$ its duration up to the beginning of trip k.

In order for the schedule to be feasible with minimum waiting time

$$\begin{cases} [\alpha^1,\ \beta^1] = [a_1 - t_{01}\ ,\ b_1 - t_{01}] \\ d^1 = t_{01} \end{cases} \tag{9}$$

For $k \geq 1$, if $\beta^k + t_{k,k+1} + d^k \leq a_{k+1}$ there is no waiting between k and k+1, and

$$\begin{cases} [\alpha^{k+1},\beta^{k+1}] = [\alpha^k,\beta^k] \cap [a_{k+1} - d^k - t_{k,k+1},\ b_{k+1} - d^k - t_{k,k+1}] \\ d^{k+1} = d^k + t_{k,k+1}\ . \end{cases} \tag{10}$$

Otherwise the minimum wait subroute is obtained with

$$\begin{cases} [\alpha^{k+1},\ \beta^{k+1}] = [\beta^k,\beta^k] \\ d^{k+1} = d^k + (a_{k+1} - b_k)\ . \end{cases} \tag{11}$$

Note that if $[\alpha^k,\ \beta^k] = \emptyset$ there is no feasible schedule for the subroute $N_r \to 1 \to \ldots \to k$. On the other hand for each time $t \in [\alpha^k,\beta^k]$, a feasible schedule with minimum waiting time can be defined for the subroute $N_r \to 1 \to \ldots \to k$ by taking for $i = 1,\ldots,k$, $t_i = t + d^i$. In addition, if the route is complete and includes v trips, arrival at the depot N_s can occur at time:

$$\begin{cases} \alpha^* = \alpha^v + d^v + t_{v0} \text{ at the earliest and} \\ \beta^* = \beta^v + d^v + t_{v0} \text{ at the latest.} \end{cases} \tag{12}$$

(t_{v0} includes the duration of trip v and the return to the depot)

7. BRANCHING TO SATISFY SCHEDULING CONSTRAINTS ON THE DIRECT ARCS

Branching is carried out when a route has no feasible schedule. For example, consider a route $N_r \to 1 \to 2 \to 3 \to 4 \to 5 \to N_s$ which is infeasible for $k \geq 4$. Belmore and Malone's [1] branching method is used. Figure 3 shows how the different ways of fixing the network variables x_{12} , x_{23} and x_{34} at 0 or 1 can be grouped into 3 branches.

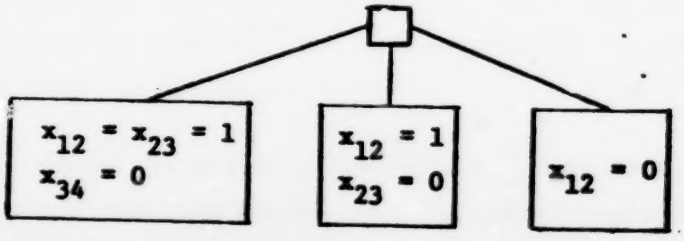

Figure 3: Branching on the arcs of the network : $1 \to 2 \to 3 \to 4$ infeasible

The problems to be handled at each branch are of the same type as the original problem but the network is modified. For example, in the first branch, arc (3,4) is dropped and nodes 1, 2 and 3 are combined while all arcs entering nodes 2 and 3 and leaving nodes 1 and 2 are also dropped.

When, for each node in the tree, each route has a non-empty starting time interval, a feasible solution to the original problem is obtained for each choice of a set of route starting times within the corresponding intervals $[\alpha, \beta]$. The number of vehicles used depends on the choice of these times and is greater than or equal to the number used by the network solution at this node.

8. DEPOT EXIT AND ENTRANCE SCHEDULE MINIMIZING THE NUMBER OF VEHICLES

To determine the number of vehicles used, the set of routes to be covered by the same vehicle must be identified. This problem is of the same type as the original problem except that the trips are replaced by routes with their corresponding time intervals $[\alpha, \beta]$. This problem is however much easier than the original because of its data structure, so the same algorithm is not used. The routes begin and end at times of the day where vehicles have time to go to the depot i.e. outside peak periods. Thus the choice of times within the intervals $[\alpha, \beta]$, and the sequencing of routes to form route blocks influences only slightly the maximum number of vehicles required at peak periods. The following method produces a solution with the minimum number of vehicles.

A lower bound $L(t)$ on the number of vehicles outside the depot at each time t is calculated. This is obtained by supposing that vehicles leave the depot as late as possible and return as early as possible i.e. they leave at time β^v and arrive at time α^*. $L(t)$ is piecewise constant and its maximum L is a lower bound on the number of vehicles necessary to cover all the routes constructed at this node of the branch-and-bound tree.

An upper bound $U(t)$ on the number of vehicles outside the depot at each time t is then estimated. This is obtained by supposing that all vehicles leave the depot as early as possible and return as late as possible for each route, i.e. they leave at time α^v and return at time β^*. The maximum U of this function is an upper bound on the number of vehicles necessary.

If $L = U$, this gives the number of vehicles required to cover all the routes independantly of their starting times. In particular, beginning all routes at starting times α^v with FIFO vehicle allocation to the routes would constitute a feasible solution with the minimum number of vehicles at this node of the branch-and-bound tree. Otherwise, a better upper bound U^1 on the minimum number of vehicles required

to cover the routes can be defined by constructing the following feasible solution. The departure times for routes leaving the depot before the peak period are set at α^v and the departure times for routes leaving after the peak period are set at β^v. Boundary cases are set arbitrarily. This bound can be improved by obtaining U^*, the minimum number of vehicles required amongst all combinations of route departure times whose intervals $[\alpha^v, \beta^v]$ or $[\alpha^*, \beta^*]$ touch the time interval for which L(t) is maximum.

In the examples tested, we always obtained $L = U$ or $L = U'$ with only one feasible solution to be constructed. In general, U^* can easily be calculated as the number of routes whose beginning or ending time intervals touch the rush hour period $\{t | L(t) = L\}$ is small. Note that this method is unsuitable for solving the original problem as several trips may begin or end in the period where L(t) is maximum, thus making combinatorial exploration very laborious.

In addition to finding the minimum number of vehicles required at this node of the branch-and-bound tree, this method identifies a feasible solution having this number of vehicles, and the same travel costs as the network solution at this node.

9. BRANCHING TO REDUCE WAITING COSTS

The exact waiting cost of this solution can be calculated as the schedule is now known. If the waiting time coincides with the lower bound used in the network relaxation we have an optimal solution for this node in the branch-and-bound tree. This was the case for all branch-and-bound nodes explored for our test problems. When the waiting time does not coincide, and the node cannot be eliminated as bounded, branching is continued on a route whose waiting time is greater than the approximation used. The same type of branching is used as that shown in Figure 3 except that an initial branch is added for which all flow variables making up the route are set at 1. For each branch, the reduced network is constructed and the upper bound on the waiting time b_{ij} is recalculated for all arcs touching the node where several trips are aggregated. The bounds are generally increased for these arcs as the aggregated node has a shorter time interval than the individual nodes composing it.

10. TESTS

The two problems of 128 and 158 trips proposed by Bokinge and Hasselstrom have in general very narrow time intervals of 0, 2 or 4 minutes with a few trips having 10-12 minute intervals. Numerical results are presented in table 1.

PROBLEM	128 TRIPS $t_i=a_i$ (3)	128 TRIPS TIME WINDOWS		158 TRIPS $t_i=a_i$ (3)	158 TRIPS TIME WINDOWS	158 TRIPS (4) TIME WINDOWS × 2	158 TRIPS (4) TIME WINDOWS × 3
Algorithm	Network	B. et H. [3]	Branch-and-bound	Network	Branch-and-bound	Branch-and-bound	Branch-and-bound
OPTIMAL SOLUTION							
- Nb. buses	34	32	32	41	41	41	40
- Empty travel time	2337	2424	2307	4695	4671	4660	4638
- Waiting time	345	152	327	447	331	292	227
- Cost (1)	5019	5000	4941	9837	9726	9612	9503 (5)
- Time (CDC173) (seconds)	10	--	5	19	20	22	24
- Nodes in branch-and-bound	--	--	5	--	1	5	23
NETWORK RELAXATION(2)							
- Nb. buses	32	--	32	41	41	41	40
- Cost	5019	--	4928	9837	9726	9582	9490

Table 1: Numerical Results

(1) Cost was defined as suggested by Bokinge and Hasselstrom:

Cost = waiting time + 2 * empty running time

(2) Network solution at first node of branch-and-bound

(3) Problems with fixed schedules at beginning of interval $[a_i,b_i]$

(4) More difficult problems constructed by multiplying the width of the time intervals by 2 or 3.

(5) This solution has not been proved optimal as the waiting time is 6 minutes over the network bound. To obtain an optimal solution, the branching described in Section 9 should be carried out.

These results show that urban transit routing problems with flexible schedules can be solved optimally. Problems with wider time intervals could be solved, and the network aggregation used allows systems of up to 500 trips to be handled without memory problems.

Savings of the order of 5% in vehicle numbers, and of 1-3% in travel and waiting times could represent appreciable economies on a large network. The savings in travel and waiting times are much higher with our exact method than with the heuristic method. The comparison is made on only one example but this would seem to be generally indicative as the example given was selected by the authors of the heuristic method.

11. REFERENCES

[1] Bellmore, Malone: Pathology of travelling salesman subtour elimination algorithm. Operations Research 19 (1971) 278-307.

[2] Bodin, Golden, Assad, Ball: Routing and Scheduling of Vehicles and Crews: The State of the Art. Computer and Operations Research 10 (1983), 69-211.

[3] Bokinge, Hasselstrom: Improved Vehicle Scheduling in Public Transport Throught Systematic Changes in the Time Table. EJOR 5 (1980) 388-395.

[4] Carpaneto, Toth: Some new branching and bounding criteria for the asymmetric travelling salesman problem. Management Science 26 (1980), 736-743.

[5] Desrosiers, Pelletier, Soumis: Plus court chemin avec contraintes d'horaires. RAIRO, Rech. Opér. 17 (1983), 1-21.

[6] Desrosiers, Soumis, Desrochers: Routing with Time Windows by Column Generation. Ecole des Hautes Etudes Commerciales de Montréal, no G83-15, presented at EURO VI, Vienna (1983).

[7] Desrosiers, Soumis: Routes sur un réseau espace-temps. Administrative Sciences Association of Canada, Man. Sci./Rech.op., vol. 3 no 2 (1982), 24-32.

[8] Desrosiers, Soumis, Desrochers, Sauvé: Routing and Scheduling with Time Windows Solved by Network Relaxation and Branch-and-bound on Time Variables. To appear in Computer Scheduling of Public Transport, vol. II, North Holland, J.-M. Rousseau Ed.

[9] Grigoriadis, Tau Hsu: RNET - The Rutgers Minimum Cost Network Flow Subroutines. Rutgers University, New Jersey, 1979.

[10] Levin: Scheduling and fleet routing models for transportation systems, Transportation Science 5, 232-255.

[11] Swersey, Ballard: Scheduling School Buses. Published at Yale School of Organization and Management and presented at TIMS/ORSA Conference, San Diego (1982).